UIUCDCS-R-77-877 UILU-ENG 77 1730

PERIPHERAL INTERCHANGE PROGRAM (PIP)

by

Douglas J. Wylie

May 1977

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN · URBANA, ILLINOIS

PERIPHERAL INTERCHANGE PROGRAM (PIP)

BY

DOUGLAS JOSEPH WYLIE

B.S., UNIVERSITY OF ILLINOIS, 1971

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 1977

Urbana, Illinois

This work was supported in part by the National Science Foundation, Grant No. NSF DCR 72-03740 A01.

TABLE OF CONTENTS

CHAPTER 1

USER'S MANUAL

1.1 INTRODUCTION

PIP (Peripheral Interchange Program) is a program to maintain the file structure of the PDP-11 Disk Operating System (DOS). PIP's functions operate on whole files, not records within the files. Some of the functions of PIP include:

Listing of directories for users
Listing of information about devices
File transfers

Commands are passed to PIP by commands typed on the keyboard. All commands are terminated by pressing the RETURN key.

1.1.1 CALLING AND EXITING

PIP is run under the program R. To execute PIP use the following sequence of commands:

.R
to which R will respond with:
?
then type PIP (RETURN)

PIP will then respond with a number sign (#). Whenever PIP prints a # it is ready to receive a command string, and after completing the command, PIP will print another #. There are two ways to exit from PIP. The first is through the /KI switch which will

return control to T. The other method is to type CTRL/C, RETURN, KI which will return you to the DOS monitor.

1.1.2 PIP SWITCHES

Most of the functions performed by PIP are specified by switches in the command string. Switches are of two types: action and qualifying. A switch is composed of a slash (/) followed by one or two letters. See Tables 1.1 and 1.2 for a summary of PIP's switches.

Call	Name	Description
/AL	Allocate	Allocate a contiguous file
/DE	Delete	Delete the file
/DI	Directory	List the directory
/EN	Enter	Enter the User Identification Code (UIC) in the Master File Directory (MFD)
/FC	Fastcopy	Copy from DECtape to DECtape
/FR	Free	List the number of free blocks and the size of the largest contiguous block
/KI	Kill	Kill PIP and return to T
/PR	Protect	Change the protection of the file
/RE	Rename	Rename the file
/UI	User Information	List number of files and number of blocks for each user entered in the MFD
/UN	Unlock	Unlock the User File Directory (UFD) to recover the file
/TD	Total Directory	List directories for all users
/WD	Withdraw	Withdraw user from the MFD

Table 1.1
Action Switches

Call	Name	Description
/BE	Before	Operate on files created before specified date
/BR	Brief Directory	List abbreviated directory
/BT	Between	Operate on files created between two specified dates
/CO	Contiguous	Force output file to be contiguous
/EX	Except	Operate on files otherwise to be rejected
/GT	Greater than	Operate on files with filenames lexicographically greater than specified filename
/IN	Inquire	Ask for user confirmation before taking action
/LE	Length	Operate on files on basis of file lengths
/LI	Linked	Force output file to be linked
/LT	Less Than	Operate on files with filenames lexicographically less than specified filename
/PA	Paginate	Print directory with headings and in pages
/PC	Protection Code	Operate on basis of protection code
/SI	Since	Operate on files created since specified date
/TY	Type	Operate on basis of file type
/UP	Update	Update file, delete if file already exists
/ZE	Zero	Zero the DECtape directory

Table 1.2
Qualifying Switches

1.2 COMMAND STRINGS

The commands processed by PIP are an extension of the DOS Command String Interpreter format. The general format of a command string is:

DEV:FILNAM.EXT[UIC]/SW1:V1:V2:...Vn/SW2:V1:V2:...

where each of the elements of the command string is explained below.

1.2.1 DEVICE SPECIFICATION

The device specification, DEV:, is a string of from one to three letters optionally followed by an octal digit. The octal digit is the unit number of the device if the system has two or more of the specified devices. If the unit number is not specified, unit 0 is assumed. If the device specification is not present, the system device is assumed.*

---

*An exception to this is the rename command (/RE) which in certain situations will assume a device other than the system device. See Section 1.5.9 for details.

1.2.2 FILENAME SPECIFICATION

The filename specification, FILNAM, consists of a letter or $ optionally followed by from one to five radix-50 characters.* PIP also employs a star (*) convention which allows a filename to be incompletely specified. When a * appears in a filename specification, the command is referring to all files which have names that match the filename up to the *. Thus:

AB*	refers to all files starting with AB
*	refers to all files.

1.2.3 FILENAME EXTENSION SPECIFICATION

The filename extension specification, .EXT, consists of a period followed by from one to three radix-50 characters. As in the filename specification, the * convention may be used to incompletely specify the filename extension.

---

*Valid radix-50 characters consist of the letters A-Z, the digits 0-9, and $.

1.2.4 USER IDENTIFICATION CODE SPECIFICATION

The user identification code specification, UIC, consists of a pair of octal numbers in the range of 0 to 176 (octal), inclusive, separated by a comma and surrounded by square brackets. The first number refers to the group and the second the user within the group. In place of either the group or user specification, * may be used to specify all groups or users. Thus:

```
[1,*]   refers to all users in group 1
[*,200] refers to all users with user number 200
[*,*]   refers to all users in all groups
```

An alternate way to specify a UIC is through predefined users. In the system file SYSHDR.STF is a list of mnemonic UIC's. Thus to specify the system user ([1,1]) it is only necessary to type: [SYS]. In the absence of an UIC specitication, the UIC of the user currently logged in on the system is assumed.

1.2.5 SWITCH SPECIFICATION

The switch specification, /SW:..., consists of a slash followed by one or two letters, and optionally followed by one or more value specifiers, each of which consists of a colon (:) followed by a string of one of more alphanumerics.

1.3 SINGLE-FILE COMMANDS

Several of the PIP commands only require one file specification. For example:

```
#DT0:ABC/DE
```

will delete the file named ABC from the DECtape mounted on unit 0. These commands are distinguished by the absence of a '<' (or '='). Single-file commands may consist of two or more file specifications in which case the action is performed on all files specified. For example:

```
#FIL.1,FIL.2/AL:10
```

will create two contiguous files of length 10 (decimal) blocks.

1.4 MULTIPLE-FILE COMMANDS

Many of the PIP commands require both a source and a destination file specification. For example:

```
#DT:FILE.NEW<FIL.OLD
```

will transfer the file FIL.OLD from the system device to the DECtape mounted on unit 0. If there is no output dataset specifier in a multi-file command (i.e., no '<' or '='), the keyboard is assumed as the output device.

1.5 ACTION SWITCHES

Action switches are used to specify the action PIP is to perform. Only one action switch is permitted in a command. When there are two or more dataset specifiers in the command (either input or a combination of input and output) the action switch may be placed after any of the dataset specifiers. Thus:

```
#LP:/DI=DT0:,DT1:
#LP:=DT0:/DI,DT1:
#LP:=DT0:,DT1:/DI
```

all mean the same: the directories for the DECtapes mounted on units 0 and 1 are to be listed on the lineprinter.

1.5.1 DIRECTORY COMMANDS

A significant number of PIP's functions deal with the listing of information about files or devices. All of these commands permit the information to be output to any device. Thus:

```
#DIRECT.FIL=/DI
```

will cause the directory for the user currently logged in on the system to be written onto the file DIRECT.FIL.

1.5.1.1 Directory

The directory switch, /DI, is used to output the directory for

the device(s) specified in the input dataset specifier(s). For each file in the directory the following information is listed:*

filename
extension
length (number of blocks)
type (either linked or contiguous)
creation date
protection code

The directory switch may be optionally followed by one or more value specifiers to indicate how the directory is to be ordered. The value may be selected from any subset of any permutation of the fields listed in Table 1.3. For example:

#/DI:C:L:E

will list the directory for the system device sorted with creation date most significant, then length, then extension. The default ordering is by extension, then by filename.

---

*If the directories are unsorted (the value specifier :U is used), the directories are not sorted by user identification code. Thus:

#/TD:U
#[*,*]/DI:U

both produce the same result.

Value Specifier	Field to Sort by
N	File Name
E	Extension
C	Creation Date
L	Length
P	Protection Code
T	File Type
U	Unsorted Directory

Table 1.3
Field Specifiers

1.5.1.2 Total Directory

The total directory switch, /TD, is used to output the directories for all the users on a device. The information listed is the same as in /DI. /TD may also have the same value specifiers as /DI. If the value specifier :U is not present the directories will be listed in order of increasing user identification codes.

1.5.1.3 User Information

The user information switch, /UI, is used to output the user identification codes, the number of files, and the number of blocks for all of the users on the device specified in the input dataset specifier. A restricted listing may be produced by either the use of a qualifying switch or by specifying a filename or extension in the input dataset specifier.*

1.5.2 Allocate

The allocate switch, /AL, is used to create contiguous files. For the allocate command, the filename and extension must be uniquely specified, i.e., the star convention may not be used. The allocate switch has for its value specifier the number (decimal) of 256 word blocks to be allocated. For example:

```
#FIL.EXT/AL:12
```

will create a contiguous file of length twelve on the system device.

1.5.3 Delete File

The delete file switch, /DE, is used to delete one or more files. For example:

```
#FILE1,FILE2/DE
```

will delete the files FILE1 and FILE2 from the system device.

---

*For example:

```
#*.COD/UI
```

will list the number of files and blocks that have the extension COD.

1.5.4 Enter User

The enter switch, /EN, is used to enter the user logged in onto the system on the device specified in the input dataset specifier.

1.5.5 Fast Copy

The fast copy switch, /FC, is used to copy DECtapes. For example:

```
#DT0:=DT1:/FC
```

will copy the information stored on the DECtape mounted on unit 1 onto the DECtape mounted on unit 0.

1.5.6 Free

The free switch, /FR, is used to list the number of free blocks and the largest group of contiguous blocks on the device specified in the input dataset specifier.

1.5.7 Kill

The kill switch, /KI, is used to exit from PIP.

1.5.8 Protection

The protection switch, /PR, is used to change the protection code of the file(s) specified in the input dataset specifier. The protection code consists of three octal digits: the first determines the access granted to the owner of the file; the second, the access granted to other users in the same group as the user; and the third, the access for all others. The protection code for the owner is composed of only two bits. Thus, the first digit should be 0, 1, 2, or 3. If the code is 1 or 3 the owner cannot write or delete the file. See Figure 1.1 for a desrciption of the protection codes. For example:

```
#FIL.EXT/PR:233
```

will change the protection code of FIL.EXT to 233.

	Function			
Code	Delete	Write	Read	Run
0	yes	yes	yes	yes
1		yes	yes	yes
2 or 3			yes	yes
4 or 5				yes
6 or 7				

Figure 1.1
Protection Codes for Group and Others

1.5.9 Rename

The rename switch, /RE, is used to rename files. If both input and output devices are specified, they must be identical. For example:

```
#DT:FIL.NEW/RE=DF:FIL.OLD
```

is illegal, whereas:

```
#DT:FIL.NEW/RE=FIL.OLD   and
#FIL.NEW/RE=DT:FIL.OLD
```

are both legal and specify the same action. Note that rename is the only command that assumes a device other than the system device. When the expanded * convention is used, rename changes the characters specified in the input dataset specifier to the characters specified in the output dataset specifier. For example:

```
#A*/RE=AB*
```

will change the name of file ABFILE to AFILE.

1.5.10 RECOVERING FILES

There are a number of infrequent ways in which a file can be left in a state which makes it inaccessible for subsequent processing. For example, if a file is open and a system crash (hardware or software) occurs causing the Monitor to be reloaded, the file will likely be left in an inaccessible state. Files which are declared inaccessible by DOS will likely have up to three things wrong with them:

1. The LOCK bit in the UFD entry for this file will be set.

2. The USAGE COUNT in the UFD entry for this file will be invalid.

3. Some blocks allocated for this file may not have been marked off in the permanent bit map.

PIP provides a partial solution to this infrequent problem with the unlock switch. The function of this switch is to restore the Lock and Usage Count fields so the file can be read. It does not make an attempt to mark any blocks in the bit map. The sequence for recovery is to use the UNLOCK switch, such as:

```
#FILE.NAM/UN
```

which allows the file to be accessed. In order to prevent the blocks from being overwritten, the file should now be transferred to another file, such as:

```
#FILE.NEW=FILE.NAM
```

which will copy the old file and guarantee that it is in the proper state. The old file should then be deleted, such as:

```
#FILE.NAM/DE
```

and the latest file can be renamed, if desired, such as:

```
#FILE.NAM/RE=FILE.NEW
```

1.5.11 Withdraw User

The withdraw user switch, /WD, is used to delete all files on the device specified in the input dataset specifier and to delete the user logged in on the system from the MFD for the device. For example:

```
#/WD
```

will withdraw the user from the system device. A special case is withdrawing the user from a DECtape. With DECtapes, there may be many users entered onto the MFD, but there is only one UFD for all the users. Thus, if there are two or more users on a DECtape, the withdraw user command clears the MFD entry for the user, but does not delete his files.

1.6 QUALIFYING SWITCHES

Qualifying switches are used for three purposes:

1. Restrict the command to a subset of the files in the input dataset specifier.
2. Specify special conditions.
3. Alter the meaning of the command.

As many qualifying switches as desired may be inserted in a command. In general, qualifying switches are only applicable to the dataset specifier immediately preceding the switch, thus in:

```
#/DT/BE:10-JUL,/SI:10-JUL
```

will produce two directories, one for all files created before July 10, and one for all files created after July 10.

1.6.1 RESTRICTING SWITCHES

The restricting switches are used to restrict the command to those files whose fields (filename, length, file type, creation date, protection code) fall within a certain range.

1.6.1.1 Restriction by Filename

The most general way to restrict by filename is via the expanded star convention (See Sections 1.2.2 and 1.2.3). In addition to the star convention, it is possible to restrict commands to all files whose names are either lexicographically less than or greater then the filename specified in the input dataset specifier.

1.6.1.1.1 Less Than

The less than or equal switch, /LT, restricts the command to all files which have filename and extension lexicographically less than or equal to that specified in the filename specification. For example:

#P*.*/LT/DI

will list a directory consisting of all files with names starting with P or less.

1.6.1.1.2 Greater Than

The greater than or equal switch, /GT, restricts the command to all files which have filename and extension lexicographically greater than or equal to that specified in the filename specification. For example:

#P*.*/GT/DI

will list a directory consisting of all files with names starting with P or greater.

1.6.1.2 Restriction by Length of File

The length restriction switch, /LE, is used to restrict the command to files whose lengths are between two values. For

example:

```
#/DI/LE:10:20
```

will list the directory of all files whose lengths are between 10 and 20 (decimal) blocks. If the first value specifier is omitted, the minimum length is taken to be zero. If the second value specifier is omitted, the maximum length is taken to be infinity. Thus:

```
#/LE::20
```

refers to all files with length less than or equal to 20 blocks, whereas:

```
#/LE:10
```

references all files of length 10 or greater.

1.6.1.3 Restriction by File Type

The file type restriction switch, /TY, is used to restrict the command to either linked files or contiguous files. For example:

```
#/DI/TY:C
```

lists a directory of all contiguous files, whereas:

```
#/DI/TY:L
```

lists a directory of all linked files.

1.6.1.4 Restriction by Creation Date

There are three ways to restrict the command by creation date:

1.	/BE	Restrict the command to files created before a specified date
2.	/SI	Restrict the command to files created after a specified date
3.	/BT	Restrict the command to files created between two specified dates

Dates may be specified in one of five ways:*

1.	DD-MMM-YY	
2.	DD-MMM	YY = current year assumed
3.	MMM-YY	DD = 1 assumed
4.	MMM	DD = 1, YY = current year assumed
5.	YY	DD = 1, MMM = JAN assumed

/BE and /SI only require one date.** For example:

```
#*.*/DE/BE:12-JUL
```

will delete all files created before July 12. /BT requires two value specifiers, neither of which may be null. For example:

```
#/DI/BT:JUN:JUL
```

will list a directory of all files created in the month of June of the current year. Note that the above is equivalent to:

```
#/DI/SI:JUN/BE:JUL
```

---

*DD = decimal day of the month
MMM = first three letters of the month's name
YY = decimal year (i.e., 75 = 1975)

**/BE and /SI may also be used without a value specifier in which case the current date is assumed.

1.6.1.5 Restriction by Protection Code

The protection code restriction switch, /PC, restricts the command to files whose protection code falls in the specified range. For example:

```
#*.*/DF/PC:200:277
```

will delete all files with protection codes between 200 and 277 (octal).

1.6.2 SPECIAL CONDITION SWITCHES

The special condition switches are used to specify additional information about the command.

1.6.2.1 Brief Directory

The brief directory switch, /BR, is used to specify the fields to be listed for directories. It is applicable in conjunction with either /DI or /TD*. /BR is followed by the fields to be listed along with the filename and extension. (See Table 1.3 for

---

*/BR may also be used as an action switch, in which case /DI is assumed.

the fields). For example:

```
#/DI/BR:C
```

will list the filename, extension, and creation date for each file, whereas:

```
#/DI/BR
```

will list only the filename and extension.

1.6.2.2 Pagination

The paginate switch, /PA, is used in conjunction with either of the directory switchs (/DI, /TD, or /BR) to provide headings for the listing of directories and to divide the listings into pages. For example:

```
#LP:=/DI/PA:50
```

will cause the line printer to form feed after every fifty (decimal) entries of the directory. If no value specifier is present, a default of sixteen is used.* If a value specifier of 0 is used, the directories will have a heading, but pagination will not occur.

---

*Sixteen was chosen since that is roughly the capacity of a CRT.

1.6.2.3 Contiguous Files

PIP normally transfers files so that the file type is preserved. The contiguous switch, /CO, may be used to force the output file to be contiguous. The switch may follow either the input or output dataset specifier. For example:

```
#ALPHA/CO=BETA  and
#ALPHA=BETA/CO
```

will both cause ALPHA to be contiguous regardless of the file type of BETA.

1.6.2.4 Linked Files

The linked switch, /LI, is used analogously to /CO. /LI forces the output file to be linked. As with /CO, /LI may follow either the input or output dataset specifier.

1.6.2.5 Updating Files

PIP normally will not permit the creation of a file with the same filename and extension as an existing file. The update switch, /UP, causes PIP to first delete the file (if it exists) before continuing with the command. As with /CO and /LI, /UP may follow either the input or output dataset specifiers.

1.6.2.6 Zeroing DECtapes

The zero switch, /ZE, is used to zero DECtapes. It initializes a DECtape with the basic file structure information required by the DOS Monitor. This switch works only for DECtapes and performs no action on other devices. It may be used in either the single or multi-file syntax. When used in the multi-file syntax, the zeroing is done before the execution of the command. For example:

```
#DT:/ZE
```

will zero DECtape mounted on unit 0, whereas:

```
#DT:/ZE=A,B
```

will first zero the DECtape and then transfer the files A and B from the system device to the DECtape. The zero switch may be optionally followed by a value specifier consisting of a decimal number. This number is the DECtape number to be assigned to the DECtape.

1.6.3 ALTERING COMMANDS

Two switches are provided to alter the meaning of the command.

1.6.3.1 Inquire

The inquire switch, /IN, is used to require user confirmation before any action is performed. For example:

```
#*.COD/DE/IN
```

will print on the keyboard the name of each file with an extension of COD followed by '?' and wait for user response before continuing with the command. The user may respond in one of three ways:

1. If the user responds with 'Y' <CR> the file will be deleted.
2. If the user responds with 'A' <CR> the whole command will be aborted.
3. If the user responds with anything other than 'Y' or 'A', the file will not be deleted but PIP will continue looking for files with extension COD.

Commands that normally require the presence of a filename in the input dataset specifier may omit the filename if /IN is used, in which case the filename and extension *.* is assumed (i.e., all files).

1.6.3.2 Except

The except switch, /EX, is used to specify that the action is to be performed on all files that otherwise would be rejected. For example:

```
#*.COD/DI/EX
```

will cause the listing of all files that do not have an extension of COD. When used in conjunction with other qualifying switches, /EX refers to all files which fail to meet any of the requirements, not only to files which fail to meet all of the requirements. Thus:

```
#/DI/BE:JUN/SI:AUG/EX
```

will cause the directory consisting of all files created before June 1 or after August 1 of the current year. Note that it does not cause the listing of the directory of all files created before June and after August.

1.7 TRANSFER COMMANDS

In the absence of an action switch, PIP assumes that the function to be performed is a file transfer. Transfers may be between file-structured or nonfile-structured devices. Transfer commands are of two types: either transferring and combining, or transferring without combining.

1.7.1 TRANSFERRING AND COMBINING

Several files may be concatenated together to form a single new file. This is done by specifying a filename in the output dataset specifier. For example:

```
#FILE.EXT=DT1:FILE.1,FILE.2
```

will combine the files FILE.1 and FILE.2 on the DECtape mounted on unit one to form the file FILE.EXT on the system device. The resultant file is always linked if two or more files are specified by the input dataset specifier(s), regardless of the file types of the input files. The contiguous switch, /CO, is ignored. Thus:

```
#FILE.NEW/CO=DT:FILE.1,FILE.2
#FILE.NEW=DT:FILE.*
#FILE.NEW=DT:FILE.EXT/GT
```

will all result in FILE.NEW being linked since in each of the above cases the possibility exists that two or more files may be combined. If there is only one input file specified, file type will be preserved. For example:

```
#FIL.NEW=DT:FIL.OLD
```

will result in FIL.NEW being contiguous if and only if FIL.OLD was contiguous. If it is desired to create a single contiguous file from two or more files, the following procedure is recommended:

```
#FIL.TMP=FIL.1,FIL.2,FIL.3
#FIL.NEW/CO=FIL.TMP
#FIL.TMP/DE
```

1.7.2 TRANSFERRING WITHOUT COMBINING

When a filename is not present in the output dataset specifier,* all the files specified by the input dataset specifier(s) are transferred as separate files. When transfers are between directory-structured devices, the file types are preserved in the absence of the link (/LI) or contiguous (/CO) switches. For example:

```
#DT1:=FIL.1, *.COD
```

will transfer FIL.1 and all files with extension COD from the system device to the DECtape mounted on unit 1, preserving their file types.

---

*The limited * convention may be used in transferring without combining. For example:

```
#AFILE.*=BFILE.*
```

will transfer all files with a filename of BFILE and change the filename to AFILE. The expanded * convention may not be used, thus:

```
#A*=B*
```

is illegal.

CHAPTER 2

PROGRAMMER'S MANUAL

The following sections deal with the algorithms in the program and how to maintain and update PIP.

2.1 OVERVIEW OF PIP

The organization of PIP is broken into three major parts. The first part is data structures used to implement the various commands. The second component is the procedures used to interpret the commands, and the third consists of the procedures used to execute the commands.

2.1.1 DATA STRUCTURES

The data structures in PIP are used to represent datasets, files, and UIC's. In addition, PIP utilizes several tables to facilitate the interpretation and execution of commands.

2.1.1.1 Representation of Datasets

Datasets are represented by the type DATASETTYPE which is declared as

```
record
        DVC,
        UNIT,
        UIC,
        NAMELEN,
        EXTLEN: integer;
        DVCSPEC: Boolean;
        NAMESPEC = (UNIQUE, NONUNIQUE, UNSPEC)
end;
```

where

DVC is the radix-50 encoding of the device name

UNIT is the unit number of the device

UIC is the specified user code (see Section 2.2.1.3)

NAMELEN is the number of characters present in the filename specification.

EXTLEN is the same as NAMELEN except it is for the filename extension

DVCSPEC is True if and only if the device was explicitly included in the dataset specifier

NAMESPEC tells how the filename and extension were specified:

UNIQUE implies name explicity specified

NONUNIQUE implies a * was found

UNSPEC implies filename not present.

2.1.1.2 Representation of Files

In order to implement all of the features of PIP, in particular the expanded * convention and the additional qualifying switches, the information about files is considered to be composed of six fields. These fields are:

1. Filename
2. Filename extension
3. Length of file
4. File type
5. Creation date
6. Protection code

When a file specification is input, the information in the dataset specifier is stored in two arrays, LOBND and UPBND which contain the minimum and maximum values respectively, for each of the above fields.* Similarly, each file in the user's UFD has its fields stored in the global array DATA. Thus, at all times, the file currently being accessed has the information about its fields available to all procedures.

2.1.1.3 REPRESENTATION OF UIC'S

PIP permits a * to be substituted for either the group or user

---

*Output files are uniquely determined so a range of values for the fields is not needed, rather a single value is sufficient to specify the output file.

number in the UIC. The UIC is stored in one word: OUTDATASET.UIC for the output dataset and INDATASET.UIC for the input dataset. To determine if a given UIC matches the UIC specifier, PIP uses the global variable MASK to mask out either the group, the user, or both. The procedure DATASETSPEC initializes MASK and DATASET.UIC according to Table 2.1.

GROUP	USER	MASK	UIC
specified	specified	-1	256*group + user
specified	unspecified	255	256*group
unspecified	specified	-256	user
unspecified	unspecified	0	0

Table 2.1
Representation of UIC

Then, a given user code (USER) is acceptable iff:

USER & MASK = UIC

The global variable USER contains the UIC of the user whose files are currently being accessed.

2.1.1.4 TABLES

PIP uses a number of tables to determine information about the given command. These tables are initialized by the procedure INITIALIZE. INITIALIZE inputs the values for the tables from the file PIP.BIN which is found in UIC [1,1]. If the file PIP.BIN does not exist, the external procedure INIT is called to create the file. A list of the tables used and their functions is found in Table 2.2.

Name	Index Set	Function
BREAKCHAR	char	Boolean array, BREAKCHAR[ch] is True iff ch is a break character (See Section 2.2.1.10)
DAYS	0..12	Integer array, DAYS[i] is the day of the year of the last day of the ith month
DEFAULTORDER	0..6	Integer array used to determine the default ordering for the listing of directories
LETTER	char	Boolean array, LETTER[ch] is True iff ch is a letter or dollar sign ($)
MULTICHAR	switches	Boolean array, MULTICHAR[sw] is True iff the switch sw needs to be specified with two characters.
OWNERCMD	switches	Boolean array, MULTICHAR[sw] is True iff the switch sw indicates a command that is restricted to the owner of the file.
RAD50	0..2,char	Integer array used to convert from ASCII to radix-50. RAD50[i,ch] is the radix-50 value for the character ch when it occurs in the ith position. RAD50[i,ch] is -1 if ch is not a valid radix-50 character.
SWITCHNAME	switches	Character array, SWITCHNAME[sw] is the ASCII representation of the switch sw.
UNRAD	0..39	Character array, UNRAD[i] is the ASCII character that in radix-50 is represented as i.
USRMUSTBETHERE	switches	Boolean array, USRMUSTBETHERE[sw] is True iff for the command determined by the switch sw, the current user must be entered onto the specified device.

Table 2.2
Tables Used by PIP

2.1.2 BASIC FLOW OF PIP

The interpretation and execution of commands is broken into four parts.

1. Determining type of command
2. Extracting the output dataset specifier
3. Checking for syntax errors in the input dataset specifier(s)
4. Extracting input dataset specifiers and executing the command for each input specifier.

2.1.2.1 Determination of Type of Command

The type of command is determined by scanning the command string and picking up action switches and any associated value specifiers. In the absence of any action switch, PIP assumes the command will be a file transfer. At the same time, it is determined whether the command is single-file or multi-file (denoted by the absence or presence of '=' or '<'in the command).

2.1.2.2 Extracting Output Dataset Specifier

If the scan determined that an output specifier was present, the dataset specifier is extracted from the command, and the file variables FILEOUT and BINARYFILE2 are assigned to the

corresponding fields of the dataset specifier. If no output specifier is present, the keyboard is assumed.

2.1.2.3 Checking for Syntax Errors

Before any action is taken, the rest of the command is checked to ensure that there are no syntax errors. If a syntax error exists, the command is aborted.

2.1.2.4 Executing Command

After it has been determined that the command is syntactically correct, each of the input dataset specifiers is again extracted from the command and the file variable INFILE is assigned to reflect the specified file. Then the actual execution of the command is begun. Section 2.2 describes the procedures used to effect the requested commands.

2.2 DESCRIPTION OF PROCEDURES

As stated earlier, the execution of a command is broken into several parts. The following procedures deal with either interpreting a command or executing a command. The procedures used for determining the syntax of the command are found in Table 2.3 and the procedures used for the execution of the command are listed in Table 2.4.

PROCEDURE	DESCRIPTION
DATASETSPEC	Extracts a dataset specifier from the command
DECIMAL	Converts ASCII string to decimal number
ENCODE	Encodes three ASCII characters into their radix-50 representation
FIELD	Determines field value specifier
FINDCOLON	Checks to see if current switch has an associated value specifier
GETMONTH	Extracts a month and returns the day of the year of the first day of the month
GETNEXTITEM	Extracts the next item from the command string
JULIAN	Converts an ASCII date to its modified Julian representation
OCTAL	Converts an ASCII string to an octal number
OCTALORSTAR	Checks the command string for either an octal number or a star (*)
SWITCH	Determines the type of switch
SYNTAXERROR	Outputs an error message

Table 2.3
Syntax Procedures

PROCEDURE	DESCRIPTION
ALLOC	Performs the DOS allocate function
ANOTHER	Searches a directory for the next entry
ANOTHERFILE	Returns the next file matching the current specifications
ASSIGN	Associates a file variable with the specified file
ASCIITRANSFER	Transfers an ASCII file
BINARYTRANSFER	Transfers a binary file
DELETECMD	Performs the delete file function
DIRECTORYCMD	Used for the listing of directory information
ENTERUIC	Enters the UIC of the current user into the MFD of the specified device
ERROR	Outputs an error message and terminates command
FASTCOPY	Copies from one DECtape to another
FINDUSER	Returns with the next user matching the current specifications
FREEBLKS	Calculates the number of free blocks on the specified device
HEADING	Outputs headings for directories
LESSTHAN	Performs an unsigned comparison
LISTENTRY	Outputs directory information about a file
LISTFILENAME	Outputs a filename
LISTFILESANDBLKS	Outputs the number of files and blocks
PRINTDVCNAME	Outputs device name
RENAMECMD	Performs the rename function
RESPONSE	Temporarily halts execution of command to seek user confirmation
SORTDIRECTORY	Sorts and outputs the directory for one user
STAT	Returns the status of a device
TRANSFERCMD	Transfers a file
UNDIV	Performs an unsigned division
UNMOD	Performs an unsigned modulus function
UNMPY	Performs an unsigned multiplication
UNPACK	Converts an integer from radix-50 to ASCII
UNPACKDATE	Converts a modified Julian date to DD-MMM-YY
VALIDFILE	Checks to see if specified filename already exists
WHOLEDIRECTORY	Sorts and outputs directories for all users on a specified device
ZEROIT	Zeroes DECtapes

Table 2.4
Execution Procedures

2.2.1 SYNTAX PROCEDURES

2.2.1.1 DATASETSETSPEC

declaration: procedure DATASETSPEC (var DATASET: DATASETYPE);

parameter: DATASET dataset to be initialized

description: DATASETSPEC extracts the next dataset specifier from the command, sets the fields of DATASET to reflect the specifier, picks up any associated qualifying switches, and returns the attributes of the specified device in the global array ATTRIBUTES.* A dataset specifier consists of four parts, any of which may be null:

1. Device specification
2. Filename specification
3. User identification specification
4. Switch list

The device specification, if present, is converted from ASCII to radix-50. The encoded device name is assigned to DATASET.DVC. If a unit is specified, DATASET.UNIT is set to the specified octal digit, otherwise, DATASET.UNIT is set to zero.

---

*See Figure 2.1 for an abbreviated description of DATASETSPEC.

```
begin
    <get next item from input>
    if ENDCHAR = ':' then
        begin
            <get device specification>
            <get next item from input>
        end;

    if ENDINDEX \= 0 then
        begin
            <get filename specification>
            if ENDCHAR \= eol then
                begin
                    <get next item>
                    <get extension specification>
                end
        end;

    if ENDCHAR = '[' then
        begin
            <get UIC specification>
        end;

    while ENDCHAR = '/' do
        begin
            <get switch>
            if <qualifying switch> then
                begin
                    <get value specifier
                     associated with switch>
                    <set field limits appropriately>
                end

            else <skip any value specifiers>
        end;

    <get attributes of device>

end;
```

Figure 2.1
Outline of DATASETSPEC

2.2.1.1.2 Filename Specification

If there is a filename specification present, the following action is taken:

1. The first three characters of its name are encoded into radix-50.*

2. If there are more than three characters, the second three are also encoded. (Any characters in excess of six are ignored).

3. If there is a filename extension in the dataset specifier, (indicated by a '.' immediately following the filename specifier) the first three characters of the extension are encoded into radix-50.

2.2.1.1.3 User Identification Specification

As explained in Section 2.1.1.3, the user identification specification may be of several forms. The global variable MASK and the UIC field of the parameter DATASET are assigned values according to Table 2.1.

An alternate way of specifying the UIC is through the use of pre-defined user names. If the user identification specification is not of the form [<octal number>, <octal number>], the characters between the brackets are encoded into radix-50. Then the array USRNAME is examined to see if a pre-defined user name has been established. If it has, DATASET.UIC is set to the corresponding user code and MASK is set to -1. If no user name

---

*See Section 2.2.1.2 for details

is found that matches the specified name, the command is terminated with an error message.

2.2.1.1.4 Switch List

The dataset specifier may be optionally followed by a list of switches. If it is, for each switch present, the following action is taken:

1. Determine the type of the switch.*
2. If it is a qualifying switch,** the action specified in Table 2.5 is performed.
3. If the switch is an action switch, the value specifiers associated with the switch are skipped.

---

*This is done with a call to SWITCH (Section 2.2.1.11).

**The action switches /TD, and /UI, both assume a user identification code specifier of [*,*], so DATASETSPEC sets MASK to -1 and INDATASETUIC to 0 to reflect this.

SWITCH	ACTION TAKEN
BE	Set upper bound for creation date to specified Julian date
BR	Get all fields to be listed for directory
BT	Set upper and lower bounds for creation dates to specifed Julian dates
CO	Set contiguous flag
EX	Set except flag
GT	Set upper bounds for filename and filename extension to -1. Note that -1 equals 65535, i.e., the largest possible number.
IN	Set the inquire flag
LE	Set the upper and lower bounds for the file length to the two specified values
LI	Set the link flag
LT	Set lower bounds for filename and filename extension to 0.
PA	Set the paginate flag and determine lines per page
PC	Set the upper and lower bounds for the protection codes to the two specified values
SI	Set the lower bound for the creation date to the specified Julian date
TY	Set the upper and lower bounds for file type so that only linked or only contiguous files will be considered
UP	Set the update flag
ZE	Set the zero flag and get DECtape number

Table 2.5
Actions Performed in Response to Qualifying Switches

2.2.1.2 ENCODE

declaration: procedure ENCODE (INDEX, K: integer);

parameters:

INDEX	index into ITEM of where encoding is to begin
K	field to be set: 0 implies first word of filename 1 implies second word of filename 2 implies filename extension

description: ENCODE converts the string of characters in ITEM pointed to by INDEX into a radix-50 packed word. The radix-50 value is stored in the array LOBND. It also sets the corresponding entry in UPBND appropriately, and the NAMESPEC field of DATASET is set to NONUNIQUE if a * is found in the ASCII string along with either the NAMELEN or EXTLEN field.

2.2.1.3 DECIMAL

declaration: function DECIMAL: integer;

description: DECIMAL returns the unsigned value of the ASCII string in ITEM.

2.2.1.4 OCTAL

declaration: function OCTAL (INDEX: integer);

parameter: INDEX index into ITEM array of where conversion is to begin

description: OCTAL converts the ASCII string pointed to be INDEX in the array ITEM to an octal number.

2.2.1.5 OCTALORSTAR

declaration: function OCTALORSTAR: integer;

description: OCTALORSTAR is used to input either the group number or user number of a UIC. It extracts the next item from the command and returns -1 it the first character of the item was a *. Otherwise OCTAL is called to get the value.

2.2.1.6 JULIAN

declaration: function JULIAN: integer;

description: JULIAN is called when PIP is expecting a date specifier. This may be from either the /BE, /SI, or /BT

switches. PIP allows six types of dates:

DD-MMM-YY	DD = day of month, MMM = first three characters of month, YY = year
DD-MMM	YY = current year assumed
MMM-YY	DD = 1 assumed
MMM	DD = 1, YY = current year assumed
YY	DD = 1, MMM = JAN assumed
<null>*	current date assumed

2.2.1.7 GETMONTH

declaration: function GETMONTH: integer;
---

description: GETMONTH returns the day of the year of the first
---

day of the month in the array ITEM.

2.2.1.8 FIELD

declaration: function FIELD: integer;

---

*A null date is valid only for /SI and /BE and is indicated by the absence of a colon after the switch.

description: FIELD extracts the next item from the command and checks the first character of the item. It returns a value according to Table 2.6.

First Character	Value Returned*	Field Specified
N	0	Filename
E	2	Filename extension
C	3	Creation date
L	4	Length
T	5	File type
P	6	Protection code
U	0	Unsorted**

Table 2.6
Field Specifiers

FIELD is used for two purposes, either to determine orderings for the listing of directories or to determine the fields to be listed in response to the /BR switch.

2.2.1.9 FINDCOLON

declaration: function FINDCOLON (SHOULDBETHERE: Boolean):

Boolean;

---

*The value returned is the index into the array DATA of the field.

**The value specifier U also sets the global variable SORT to False.

parameter: SHOULDBETHERE True if the switch requires a value specifier

description: FINDCOLON checks to see if the current item of the command was terminated by a ':'. If so, the next item of the command string is extracted and FINDCOLON returns True. If it was not terminated by a ':', SHOULDBETHERE is checked and if SHOULDBETHERE is True, FINDCOLON terminates the command with an error message. If SHOULDBETHERE is False, FINDCOLON returns False.

2.2.1.10 GETNEXTITEM

declaration: procedure GETNEXTITEM;

description: GETNEXTITEM extracts the next item from the command and stores it in the array ITEM. An item is defined as any (possibly null) string of alphanumerics terminated by one of:

: . , - = < [ ] eol

In addition to returning the item, the following globals are also set by GETNEXTITEM:

FIRSTCHAR	First character of item
ENDCHAR	Character that terminated item
ENDINDEX	Number of characters in item
POSITION	Array the contains the position in the command of each character in the item

2.2.1.11 SWITCH

declaration: function SWITCH: SWITCHTYPE;

description: SWITCH extracts the next item from the command and then checks to see if it matches any of the switches in the array SWITCHNAME. If a match is found, SWITCH returns the corresponding switch. If no match is found, an error message is printed and the command is aborted.

2.2.1.12 SYNTAXERROR

declaration: procedure SYNTAXERROR (MESSAGE: text;
INDEX: integer);

parameters: MESSAGE Error message to be printed on keyboard
INDEX Index into ITEM of offending character

description: SYNTAXERROR outputs the error message under the character where PIP got confused. The command is then aborted.

2.2.2 EXECUTION PROCEDURES

2.2.2.1 ENTERUIC

declaration: procedure ENTERUIC;

description: ENTERUIC enters the UIC of the user currently logged in on the system into the MFD for the device on which the file INFILE resides.* First the first MFD block is read into the array MFD, then the second MFD block is input and ENTERUIC looks for an empty entry (one whose UIC is zero). If such an entry is found, the UIC of the current user is inserted into this position. The pointer to the UFD block is then cleared** and the MFD block is rewritten onto the device. Currently, PIP will not attempt to extend the MFD, so if there are no free entries, an error message is printed.

---

*Although ENTERUIC is normally called in response to the /EN switch, it may also be invoked (after user confirmation) when the UIC specified in the output specifier is not found in the MFD for the device.

**If the device is a DECtape, the UFD pointer is not cleared, rather it is set to 102 (octal) since 102 is the start address for all UFD's of a DECtape.

2.2.2.2 TRANSFERCMD

declaration: procedure TRANSFFRCMD;

description: TRANSFERCMD transfers the files specifed by the current input dataset specfier to the current output specifier. For each file that matches the current specifications TRANSFER either performs an ASCII or binary transfer. A binary transfer is used if both input and output devices support binary, otherwise the file is transferred as an ASCII file.

2.2.2.3 ASCIITRANSFER

declaration: procedure ASCIITRANSFER;

description: ASCIITRANSFER transfers an ASCII file from the current input device to the current output device. The transfer is done one character at a time.

2.2.2.4 BINARYTRANSFER

declaration: procedure BINARYTRANSFER;

description: BINARYTRANSFER transfers a file from a device that supports binary input to a device that supports binary output. As such, it handles all transfers between file-structure devices. Since PIP endeavors to preserve file type when transferring files and since there are two types of transfer commands (transfer with combining and transfer without combining), BINARYTRANSFER must do the following if the output device is file-structured:

1. If the transfer is to be without combining (OUTDATASET.NAMFSPEC \= UNIQUE) then reassign the output file (forcing a close on the previous file).

2. Check to see that there is not already a file in the UFD with the same filename and extension. (See Section 2.2.2.15 for details).

Then, (whether transfering with or without combining) if the input file was contiguous, or the contiguous flag (CONTIG) is True, allocate contiguous space for the output file (unless the linked flag (LINK) is True.* After the file has been allocated (if required) then the file transfer is done. The actual transfer itself is done differently for linked and contiguous

---

*The link flag may be set explicitly by the user with the /LI switch, or PIP may determine that the transfer must be linked. PIP forces the file to be linked if:

1. The input device is non-file structured;
2. More than one input data set specifier is present, or;
3. A * was used in the specification of the input data set filename.

files. If the input is contiguous then the file must be transferred in 256 word blocks, whereas, if the file is linked, the transfer must be in blocks of 255 words.*

2.2.2.5 DIRECTORYCOMMAND

declaration: procedure DIRECTORYCOMMAND

description: DIRECTORYCOMMAND implements the /TD, /DI, and /UI functions. Depending on the command and whether or not the sort flag is set (SORT), a decision is made as to the action to be performed. If the command is UI, then WHOLEDIRECTORY is called. Otherwise:

```
if SOPT then
   if CMD = TD then WHOLEDIRECTORY
   else
      repeat
         SOPTDIRECTORY
      until \FINDUSER
else
   repeat
      while ANOTHERFILE do
         LISTENTRY
   until \FINDUSER
```

*Due to peculiarities in DOS, the last block in a linked file cannot be transferred as 255 words. Instead, it must be transferred as 254 words plus one byte.

2.2.2.6 WHOLEDIRECTORY

declaration: procedure WHOLEDIRECTORY;

description: WHOLEDIRECTORY sorts the users found on the input device and then does one of two things:

1. If the command is TD, SORTDIRECTORY is called to output the directory for each user.
2. If the command is UI, the UIC's and number of files and number of blocks in the user's UFD is listed.

2.2.2.7 SORTDIRECTORY

declaration: procedure SORTDIRECTORY

description: SORTDIRECTORY sorts and outputs the directory for a single user. A doubly-linked list insertion sort is used.

2.2.2.8 LISTENTRY

declaration: procedure LISTENTRY;

description: LISTENTRY outputs the directory entry for the file whose fields are stored in the global array DATA. If the paginate flag (PAGINATE) is set, LISTENTRY outputs a formfeed and prints headings when the end of a page is reached. The global array LIST is inspected to see which fields of the entry are to be output.

2.2.2.9 LISTFILESANDBLKS

declaration: procedure LISTFILESANDBLKS;

description: LISTFILESANDBLKS is called by the various directory procedures to output the number of files and blocks found. If the output device is the keyboard and no files were found, then the output is suppressed.

2.2.2.10 PRINTDVCNAME

description: PRINTDVCNAME outputs the device name of the current input device (INDATASET.DVC), along with its unit number. If the device is a DECtape, then instead of the unit number of the device, the number of the DECtape is output (unless the DECtape is not numbered).

2.2.2.11 PRINTIT

declaration: procedure PRINTIT (name: text;
VAL: integer);

parameters: NAME identifier to be output
VAL value to be output

description: PRINTIT outputs VAL followed by its description and optionally followed by an 'S' (the 'S' is printed if VAL \= 1).

2.2.2.12 PRINTUIC

declaration: procedure PRINTUIC;

description: PRINTUIC outputs the UIC of the current user.

2.2.2.13 HEADING

declaration: procedure HEADING;
---

description: HEADING is called to print the headings for the
---

listing of directories.

2.2.2.14 FREEBLKS

declaration: procedure FREEBLKS;
---

description: FREEBLKS checks the bit maps for the input device
---

and outputs the number of free blocks and the largest group of contiguous blocks on the device. Prior to being called, the array MFD must contain the first MFD block for the device.

2.2.2.15 ANOTHER

declaration: function ANOTHER (var DIRECTORY: array [0..255]
---
of integer;
INCREMENT, MAXINDEX: integer;
var ADDR, INDEX: integer;
var CHANGED; Boolean) Boolean;

parameters:

DIRECTORY	Directory to be searched
ADDR	Device address of current directory block
INDEX	Index into directory of where search is to begin
INCREMENT	Number of words per entry
MAXINDEX	Index of last entry in block
CHANGED	True if current directory block has been modified.

description: ANOTHER searches through DIRECTORY looking for a
---

non-zero entry. The search begins at INDEX into the current directory block. If such an entry is found, ANOTHER returns True and INDEX points to the next entry after the entry found. ANOTHER is used to search either a MFD or UFD. If the end of the current directory block is reached before a non-zero entry is found, ANOTHER does the following:

1. If CHANGED is True, the current directory block is rewritten to the device.

2. If there is another block associated with the directory (its link word is not zero), the next directory block is input, INDEX is set to 1, ADDR is set to the address of the next block, and the search continues.

3. If there is not another block associated with the directory (its link word is zero), then ANOTHER returns False.

2.2.2.16 ANOTHERFILE

declaration: function ANOTHERFILE: Boolean;

description: ANOTHERFILE checks to see if there is another file in the current user's UFD that matches the current input dataset specifier. If such a file is found, the following globals are assigned:

N1	First word of filename
N2	Second word of filename
EXT	Filename extension
DATA	Array that contains the values of each of the fields of the file.

ANOTHERFILE calls ANOTHER to get the next non-zero entry of the UFD and then checks to see if all of its fields fall within the ranges of the current input dataset specifier. (The arrays LOBND and UPBND contain the minimum and maximum values, respectively, for each of the fields.) After it has been determined whether or not the file is in range, the except flag (EXCEPT) is checked. If EXCEPT is True, then the file is in range if and only if it falls outside of at least one of the fields. If, at this point, the file is found to be acceptable, the inquire flag (INQUIRE) is checked. If it is True, then user confirmation is solicited before accepting the file. This process is continued until either a file is found that is acceptable (in which case ANOTHERFILE returns True) or there are no more files in the user's UFD (ANOTHERFILE returns False).

2.2.2.17 FINDUSER

declaration: function FINDUSER: Boolean;

description: FINDUSER checks the MFD for the current input device* and returns True if a UIC is found which matches the UIC specifier. The procedure DATASETSPEC (see Section 2.2.1.1) constructs a mask (MASK) to applied to all UIC's to determine if they match the input dataset specifier. FINDUSER also sets the global USER to the UIC for the user if a user is found with the required attributes.

2.2.2.18 RESPONSE

declaration: function RESPONSE: Boolean;

description: RESPONSE is used to seek user confirmation before

---

*FINDUSER is also used to determine if the UIC of the output dataset specifier is on the output device.

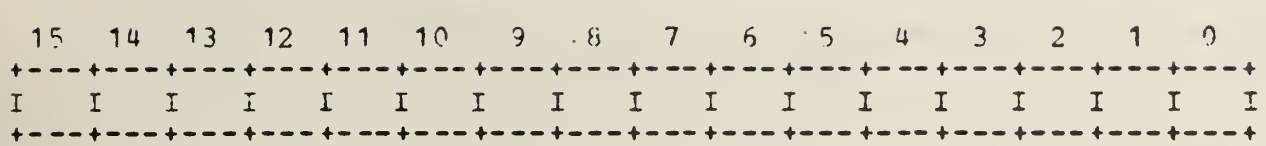
continuing the execution of a command. If the user types 'Y' <CR>, RESPONSE returns True. If the user types 'A' <CR>, the whole command is aborted. Any other response causes RESPONSE to return False.

2.2.2.19 STAT

declaration: procedure STAT (FIL: file of character; var DVCNAME: integer);

parameters:
- FIL — File whose device status is to be determined
- DVCNAME — Standard device name of device

description: STAT returns the characteristics of the device on which the file FIL resides in the global array ATTRIBUTES. The parameter DVCNAME is also set to the standard device name for the device (in radix-50). Table 2.7 lists the meanings of the items in ATTRIBUTES.



Device Status Word

Bit Position*	One Implies
0	device will support multidataset activity
1	device will handle output
2	device will handle input
3	device will handle binary data
4	device will handle ASCII data
5	driver has a special function entry
6	driver has a CLOSE entry
7	driver has an OPEN entry
8	device is a terminal
9-13	spares (not used)
14	device is DECtape
15	device is directory structured

Table 2.7
Device Status Word

2.2.2.20 ALLOC

declaration: procedure ALLOC (F: file of character; BLKS:
integer);

parameters: F Contiguous file to be created

BLKS Length of file (number of blocks)

description: ALLOC performs the file allocate function. Prior to invocation, the file block and link block associated with the

---

*The bit position is also the index into the Boolean array ATTRIBUTES. Thus, ATTRIBUTES[14] = True implies that the device in question is a DECtape.

file must be inititialized via a call to ASSIGN.

2.2.2.21 ASSIGN

declaration: procedure ASSIGN (var F: file of character; DVC,
UNIT,
NAME1, NAME2, EXT, UIC: integer);

parameters:

F	File whose link blocks and file blocks are to be initialized
DVC	Device where file resides (radix-50)
UNIT	Unit number of device
NAME1	First word of filename (radix-50)
NAME2	Second word of filename (radix-50)
EXT	Filename extension (radix-50)
UIC	UIC of owner of file

description: ASSIGN initializes the file block and link block for the file F. In the current implementation of PASCAL, this information is stored as shown in Figure 2.2. If DVC is zero, then ASSIGN does not modify the link block. If UIC is zero, ASSIGN does not change the user ID code of the file block.

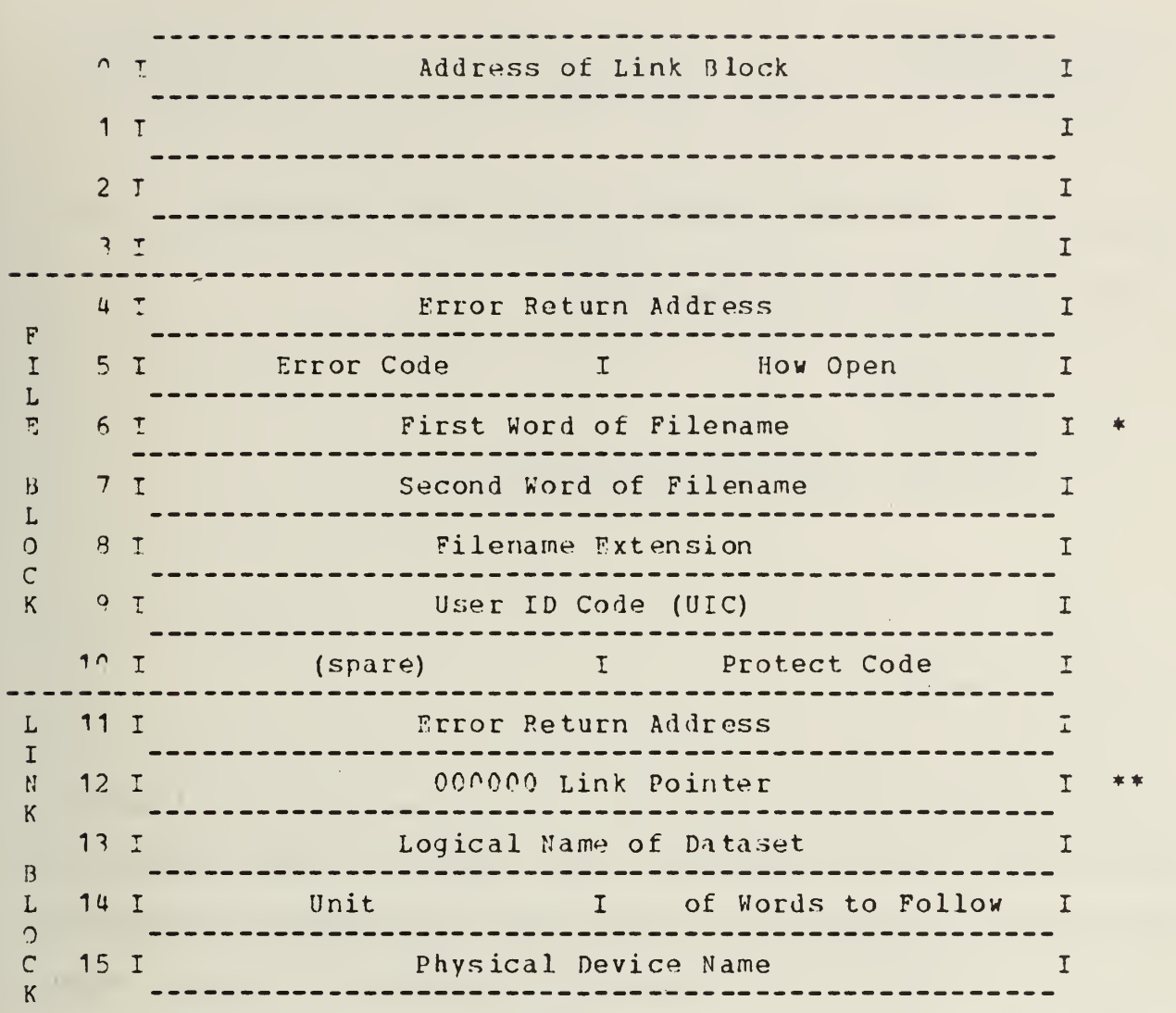


Figure 2.2
File Block and Link Block

---

* Address of file block, See Figure 2-6, page 2-63 DOS Manual
** Address of link block, See Figure 2-5, page 2-61 DOS Manual

2.2.2.22 RENAMECMD

description: RENAMECMD performs the rename function. This function is divided into five parts:

1. Masking
2. Shifting
3. Adding
4. Changing Extension
5. Checking validity of new name

2.2.2.22.1 Masking

The masking operation is done with the unsigned modulus function (UNMOD, see Section 2.2.2.31 for details). The number of characters to be masked out is the number of characters specified by the current inputdataset specifier. This value (INDATASET.NAMELEN) is determined by the procedure DATSETSPEC) (Section 2.2.1.1) (See Table 2.8)

Number of Characters to be Masked	First Word of Name (N1)	Second Word of Name (N2)
0	-	-
1	N1 mod 1600	-
2	N1 mod 40	-
3	0	-
4	0	N2 mod 1600
5	0	N2 mod 40
6	0	0

Table 2.8
Masking Filename

2.2.2.22.2 Shifting

After the characters have been masked out of the old filename, the remaining characters will have to be shifted appropriately. The number of places to shift is determined by the difference in the number of characters specified in the input and output dataset filename specifiers (See Tables 2.9 and 2.10).

Number of Positions to be Shifted Left	First Word of Name (N1)	Second Word of Name (N2)
1	N1*40+N2/1600	40*(N2 mod 1600)
2	N1*1600+N2/40	1600*(N2 mod 40)
3	N2	0
4	40*(N2 mod 1600)	0
5	1600*(N2 mod 40)	0

Table 2.9
Shifting Filename Left

Number of Positions to be Shifted Right	First Word of Name (N1)	Second Word of Name (N2)
1	N1/40	1600*(N1 mod 40)+N2/40
2	N1/1600	40*(N1 mod 1600)+N2/1600
3	0	N1
4	0	N1/40
5	0	N1/1600
6	0	0

Table 2.10
Shifting Filename Right

2.2.2.22.3 Adding in New Characters

To complete the changing of the filename, the radix-50 value for the new characters (specified in the output filename specifier) is added into the resultant name.

2.2.2.22.4 Changing Extension

The extension is modified in a manner analogous to the process for the filename. The process is somewhat simpler since the extension is only one word long.

2.2.2.22.5 Checking Validity of New Name

When the new filename and extension have been determined, the new

filename is checked to ensure that it is valid filename* and the UFD for the user is inspected to ensure that the new filename is not already there.

2.2.2.23 FASTCOPY

declaration: procedure FASTCOPY;

description: FASTCOPY implements the copying of one DECtape to another. The copying is done in two passes: first forward and then backwards with calls to the procedure TRAN (see Section 2.2.2.29).

2.2.2.24 LISTFILENAME

---

*Illegal filenames can be of two types: either the filename is null or it starts with a digit. For example, the command:

```
*/RE=A*
```

will produce a null filename when applied to the file A and an illegal filename when applied to the file A1.

declaration: procedure LISTFILENAME (NAME1, NAME2, EXT:
integer): integer;

parameters:
NAME1 First word of filename (radix-50)
NAME2 Second word of name (radix-50)
EXT Filename extension (radix-50)

description: LISTFILENAME outputs the name of the file and its extension to the keyboard.

2.2.2.25 GETUIC

declaration: function GETUIC: integer;

description: GETUIC returns the UIC of the user currently logged in on the system.

2.2.2.26 LESSTHAN

declaration: function LESSTHAN (A, B: integer); Boolean;

parameters:
A
B Numbers to be compared

description: LESSTHAN performs an unsigned comparison between A
---

and B. It returns True if A is less than B.

2.2.2.27 ERROR

declaration: procedure ERROR (MESSAGE: text);
---

parameter: MESSAGE Error message to be printed
---

description: ERROR outputs the error message passed to it and
---

aborts the command by doing an EXIT.

2.2.2.28 TODAY

declaration: function TODAY: integer;
---

description: TODAY returns the current date in modified Julian.

The modified Julian date is: (Year - 1970) * 1000 + day of year.

2.2.2.29 TRAN

declaration: procedure TRAN (BLOCK: integer;
---
var BUFFER: array [0..bufsize] of integer;
F: file of character; DIRECT: integer);

parameters:

BLOCK Device block number

BUFFER Array to be read or written

F File variable on device where TRAN is to be performed

DIRECT Direction of transfer

description: TRAN performs TRAN level requests. The direction
---

of the transfer is determied by DIRECT. The possible values of DIRECT are:

2	Write
4	Read
2050*	Reverse write
2052*	Reverse read

2.2.2.30 UNDIV

declaration: function UNDIV (A, B): integer;
---

parameters:

A Dividend

B Divisor

description: UNDIV returns the quotient of A divided by B where
---

A is taken as an unsigned number and B is either 40 or 1600

---

*Applicable to DECtape only.

(decimal).

2.2.2.31 UNMOD

declaration: function UNMOD: (A,B: integer): integer;

parameters: A

B

description: UNMOD performs the modulus function where A is an unsigned number and B is 40 or 1600 (decimal). If A is positive, UNMOD returns A mod B. If A is negative, UNMOD returns (1536 + A) mod B. When A is negative, the unsigned value of A is 65536 + A. Therefore:

val (A) = 65536 + A

val (A) = 64000 + 1536 + A

Thus for B = 40:

A mod 40 = (64000 + 1536 + A) mod 40

A mod 40 = (64000 mod 40 + [1536 + A] mod 40) mod 40

A mod 40 = (0 + [1536 + A] mod 40) mod 40

A mod 40 = (1536 + A) mod 40

Similary for B = 1600:

A mod 1600 = (64000 + 1536 + A) mod 1600

A mod 1600 = (6400 mod 1600 + [1536 + A] mod 1600) mod 1600

A mod 1600 = (C + [1536 + A] mod 1600) mod 1600

A mod 1600 = (1536 + A) mod 1600

2.2.2.32 UNMPY

declaration: function: UNMPY (A, B): integer;

parameters: A Multiplicand

B Multiplier

description: UNMPY returns the unsigned product of A and B, where B is either 40 or 1600.

2.2.2.33 UNPACK

declaration: procedure UNPACK (var F: file of character;
VAL: integer);

description: UNPACK converts the radix-50 string in VAL into ASCII and outputs it to the file F.

2.2.2.34 UNPACKDATE

declaration: procedure UNPACKDATE (JULIAN: integer);

parameter: JULIAN Date (modified Julian)

description: UNPACKDATE converts JULIAN to the date it represents (DD-MMM-YY) and outputs it to the current output file.

2.2.2.35 VALIDFILE

declaration: function: VALIDFILE (NAME1, NAME2, EXT: integer): Boolean;

parameters: NAME1 First word of filename

NAME2 Second word of filename
EXT filename extension

description: VALIDFILE checks to see if there is a file with the given name and extension in the current output user's UFD. If there is not, (eof is True), VALIDFILE returns True. If the file already exists (eof is False) then one of three things is done:

1. If the update flag is set (UPDATE) the file is deleted and VALIDFILE returns True.
2. If the output UIC is not that of the user currently logged in, an error message is printed and the command is aborted.
3. If the output UIC is that of the user logged in,

the user is asked if he wants to update, i.e., rewrite the file. If the user responds with 'Y', the file is deleted and VALIDFILE returns True. If the user responds with anything other than 'Y' (except 'A', see Section 2.2.2.18) VALIDFILE returns False.

2.3 MAINTAINING PIP

The preceding sections describe the procedures and data structures utilized by PIP. The program may be modified by adding switches. Of the two types of switches, qualifying switches are most easily added.

2.3.1 QUALIFYING SWITCHES

If addition qualifying switches are desired, the following steps are necessary:

1. Change the type SWITCHTYPE to include the new switch.
2. Modify the external procedure INIT to initialize the switch's name.
3. Then, if the switch is to be used to restrict files, determine which of the fields of the file are affected and change DATASETSPEC to set the appropriate elements of LOBND and UPBND.

2.3.2 ACTION SWITCHES

If additional commands are to be added to PIP the following steps are required*

---

*The only changes necessary would be in the procedure DATASETSPEC.

1. Change the type SWITCHTYPE to include the new switch

2. Modify INIT in initialize the name of the switch and to set the arrays OWNERCMD and USRMUSTBETHERE.

3. Write the required procedures to implement the command.

2.3.3 POSSIBLE EXTENSIONS

The number of possible extensions is practically unlimited. Three possible extension are given below.

2.3.3.1 MODIFICATION OF ENTER COMMAND

One useful extension of PIP would be to have an optional value specifier with the /EN switch consisting of the name by which the user wishes to be identified. These pre-defined UIC's are currently stored in the file SYSHDR.STF.

2.3.3.2 MODIFICATION OF THE /LT AND /GT SWITCHES

It might be useful to have value specifiers associated with these switches, i.e.,

/GT:A.B/LT:X.Y/DI

might be a possible syntax to specify all files with filenames between A.B and X.Y. The current data structure of PIP would support such an extension without any modifications.

2.3.3.3 Binary Output of Files

In some instances, the user may wish to examine a binary file either in binary or in octal. The addition of the switch /BI for binary output to either the line printer or keyboard would facilitate this action.

BIBLIOGRAPHY

Digital Equipment Corporation, DOS/BATCH Software Support Manual, Order number DEC-11-OSSMA-D, 1974.

Digital Fquipment Corporation, MACRO-11 Assembler Programmer's Manual Order number DEC-11-OMACA-A-D, June 1972.

Digital Equipment Corporation, PDP-11 Disk Operating System Monitor Programmer's Handbook, Order number DEC-11-OMONA-A-D, October 1972.

Digital Equipment Corporation, PDP-11 File Utility Package (PIP) for the Disk Operating System, Order number DEC-11-PIDB-D, September 1971.

BIBLIOGRAPHIC DATA SHEET	1. Report No. UIUCDCS-R-77-877	2.	3. Recipient's Accession No.

4. Title and Subtitle

Peripheral Interchange Program (PIP)

5. Report Date: May 1977

6.

7. Author(s): Douglas J. Wylie

8. Performing Organization Rept. No.

9. Performing Organization Name and Address

Department of Computer Science
University of Illinois
Urbana, Illinois 61801

10. Project/Task/Work Unit No.

11. Contract/Grant No.: NSF DCR 72-03740 A01

12. Sponsoring Organization Name and Address

National Science Foundation
Washington, DC

13. Type of Report & Period Covered

14.

15. Supplementary Notes

16. Abstracts

PIP is a program which maintains the file structure of the PDP-11 Disk Operating System. The version of PIP described in this document is written in Pascal-11, thus permitting the quick and simple implementation of many expanded features. These include sorted directories, a greatly expanded switch facility, and greater flexibility in specifying file names.

17. Key Words and Document Analysis. 17a. Descriptors

PDP-11
Pascal-11
PIP
file handling

17b. Identifiers/Open-Ended Terms

17c. COSATI Field/Group

18. Availability Statement	19. Security Class (This Report) UNCLASSIFIED	21. No. of Pages
	20. Security Class (This Page UNCLASSIFIED	22. Price